

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Автоматики та управління в технічних системах**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ О.І. Ролік

«\_\_\_» \_\_\_\_\_ 20\_\_р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
з напрямку підготовки 151 «Автоматизація та  
комп'ютерно-інтегровані технології»  
на тему «Телеграм бот для дистанційного навчання»**

Виконав:

Студент IV курсу, групи ІА-62

Юхновець Антон Дмитрович \_\_\_\_\_

Керівник:

кандидат технічних наук, доцент

Савчук Олена Володимирівна \_\_\_\_\_

Рецензент

Доктор технічних наук, професор кафедри ОТ

Сімоненко Валерій Павлович \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2020

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Автоматики та управління в технічних системах**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

«Затверджую»

Завідувач кафедри

\_\_\_\_\_ О.І. Ролік

«\_\_\_» \_\_\_\_\_ 20\_\_р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Юхновцю Антону Дмитровичу**

1. Тема проєкту «Телеграм бот для дистанційного навчання», керівник проєкту Савчук Олена Володимирівна к.т.н., доцент, затверджені наказом по університету від «\_\_\_» \_\_\_\_\_ 20\_\_р. № \_\_\_\_\_

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту: Телеграм бот для дистанційного навчання.

4. Зміст пояснювальної записки: опис об'єкта проєктування, аналіз вимог, аналіз предметної області, огляд існуючих рішень. Розробка Телеграм бота, опис власної реалізації та тестування бота.

5. Перелік графічного матеріалу: схема зв'язків баз даних, алгоритм роботи програми, діаграма розгортання, діаграма маршрутизації процесів.

6. Консультанти розділів проєкту

Розділ	Прізвище, ім'я, по батькові консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

### Календарний план

№	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Аналіз предметної області	14.04.2020	
2.	Огляд існуючих рішень	19.04.2020	
3.	Створення діаграми зв'язків БД	27.04.2020	
4.	Створення алгоритму роботи	4.05.2020	
6.	Розробка бази даних, сервера, бота	10.05.2020	
7.	Проведення тестування	1.06.2020	
8.	Створення діаграми розгортання	3.06.2020	
9.	Розробка діаграми маршрутизації процесів	6.06.2020	

Студент

А.Д. Юхновець

Керівник

О.В. Савчук

## АНОТАЦІЯ

Юхновець А.Д. Телеграм бот для дистанційного навчання. КПІ ім. Ігоря Сікорського, Київ, 2020.

Проект містить 60 с. тексту, 14 рисунків, 23 таблиці, посилання на 15 літературні джерела та 4 кресленника.

Об'єктом дослідження є бот для проведення дистанційного навчання.

Метою дипломного проекту є створення інструменту, що автоматизує взаємодію між користувачем та платформою і спрощує проведення навчання студентів у дистанційному режимі.

У дипломному проекті розроблено бот, що автоматизує роботу користувача з Телеграм для отримання необхідної інформації з навчального процесу. Розроблено алгоритм роботи, базу даних та зв'язок у базі. Проведений аналіз вимог, огляд існуючих рішень та проектування. Проведен аналіз інструментів створення бота, а також можливостей Телеграм як платформи для реалізації поставлених цілей. Проведений опис процесу розробки за допомогою мови програмування JavaScript, створення бази даних із використанням SQL та серверної частини використовуючи Express.js та Node.js. Проведено тестування створеного інструмента.

Отримані результати можуть бути корисними при необхідності проведення освіти студентів без можливості безпосереднього контакту із ними.

**КЛЮЧОВІ СЛОВА:** БОТ, МЕССЕНДЖЕР, ТЕЛЕГРАМ, ЧАТ-БОТ, ДИСТАНЦІЙНЕ НАВЧАННЯ.

## ANNOTATION

Yukhnovets A.D. Telegram bot for distance learning. Igor Sikorsky Kyiv Polytechnic institute, Kyiv, 2020.

The project contains 60 pages of text, 14 figures, 23 tables, links to 15 literary sources and 4 design documents.

The object of research is a bot for distance learning.

The purpose of the research is creating a tool that automates the interaction between the user and the platform and simplifies the training of students at a distance mode.

In the diploma project is developed the bot which automates work of the user with Telegrams for reception of the necessary information on educational process is. An algorithm, database and communication in the database have been developed. Requirements analysis, review of existing solutions and design. The analysis of tools of creation of a bot, and also possibilities of Telegrams as a platform for realization of the set purposes is carried out. A description of the development process using the JavaScript programming language, creating a database using SQL and the server part using Express.js and Node.js. The created tool was tested.

The obtained results can be useful if students need to be educated without the possibility of direct contact with them.

**KEY WORDS: BOT, MESSENGER, TELEGRAM, CHAT-BOT, DISTANCE LEARNING.**

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка	
1			Документація загальна				
2							
3			Знову розроблена				
4							
5	A4	ІА62.320БАК.002 ПЗ	Пояснювальна записка	66			
6							
7	A3	ІА62.320БАК.003 Д1	Схема зв'язків баз даних	1			
8							
9	A3	ІА62.320БАК.004 Д2	Алгоритм роботи програми	1			
10							
11	A3	ІА62.320БАК.005 Д3	Діаграма розгортання	1			
12							
13	A3	ІА62.320БАК.006 Д4	Схема маршрутизації процесів	1			
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
Зм.	Аркуш	№ докум.	Підпис	Дата			
Розроб.	Юхновець						
Перевір.	Савчук						
Реценз.							
Н. Контр.	Шинкевич						
Затв.					Літ.	Аркуш	Аркушів
					Т	1	1

## ЗМІСТ

ЗМІСТ .....	2
ВСТУП .....	4
1 ОПИС ОБ'ЄКТА ПРОЄКТУВАННЯ .....	6
2 АНАЛІЗ ВИМОГ .....	8
3 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
3.1 Переваги Телеграма як платформи .....	10
3.2 Телеграм боти.....	11
3.3 Аналіз інструментів створення бота .....	13
3.4 Висновки до розділу .....	15
4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	16
4.1 Найпоширеніші платформи для дистанційного навчання.....	16
4.2 Zoom .....	16
4.3 Moodle .....	17
4.4 Discord.....	19
4.5 Висновки до розділу .....	21
5 РОЗРОБКА TELEGRAM-БОТА.....	22
5.1 Реєстрація чат-бота та отримання токена.....	22
5.2 Середовище розробки.....	24
5.3 Модуль Telebot.....	25
5.3.1 Події TeleBot .....	26
5.3.2 Модифікатори TeleBot .....	27
5.3.3 Методи TeleBot .....	28
5.4 Розробка веб-серверу.....	29
5.4.1 HTTP-запити .....	29
5.4.2 Express.js .....	30
5.5 Розробка бази даних .....	34
5.5.1 Відношення баз даних.....	34
5.5.2 MySQL та SQL .....	38

Зм.	Аркуш	№ докум.	Підпис	Дата								
Розроб.		Юхновець				Літ.		Аркуш		Аркушів		
Перевір.		Савчук				Т			1		1	
Реценз.												
Н. Контр.		Шинкевич										
Затв.												





## ВСТУП

За останні роки у світі дуже швидко змінюються умови та стандарти в різних галузях. Індустріалізація та впровадження нових технологій у повсякденне життя змушують компанії та підприємства впроваджувати нові заходи для оптимізації робочого процесу, а також постійно проводити аналіз вже існуючих методів та рішень, щоб не відставати від трендів та потреб сучасного суспільства. Гостро стало питання адаптації вже існуючих підходів навчання співробітників, а також освіти під тенденції розвитку нових технологій.

В останні роки було створено безліч платформ та інструментів для забезпечення освітою студентів. Деякі з них активно використовуються у багатьох країнах вже не один рік і досягли певного успіху у цій галузі. Завдяки ним освіту можуть отримати студенти з усіх кінців світу без потреби знаходитись в навчальному закладі. Такий підхід найбільш підходить під умови глобалізації, що створені сьогоднішніми реаліями життя.

За допомогою ІТ-технологій також розвинулась соціальна складова в суспільстві. Люди мають можливість кожен день спілкуватися через соціальні мережі, а останнім часом особливої популярності набрали месенджери через простоту і доступність. Одним з таких месенджерів є Telegram, за допомогою якого реалізується метод навчання, розроблений в даній роботі.

Telegram - додаток-месенджер, розроблений на C ++ Павлом Дуровим. Від інших месенджерів (WhatsApp, Viber, Messenger) має перевагу в швидкості, а також конфіденційності користувачів. Довгий час телеграм не мав локалізації для країн СНД, через це месенджер користувався популярністю, в більшості, у ІТ-фахівців і зарубіжних країн - Італії, Іспанії і Бразилії.

Крім звичайних користувачів телеграм, в месенджері присутні боти. Бот – це програма, що автоматично виконує заздалегідь запрограмовані або задані іншим методом дії за допомогою інтерфейсу платформи, на основі якої реалізований бот. Боти можуть виконувати різні функції, але основна - відповідати користувачу на

Зм.	Лист	№ докум.	Підпис	Дата		Лист

запити. Функціонал бота може бути обмежений мовою програмування або іншого інструменту створення бота, а також обраної платформи для його розробки, як Телеграм. Причина, з якої для розробки бота був обраний телеграм - високий функціонал месенджера. Телеграм має безліч передових функцій, якими не володіють його аналоги, наприклад, створення опитувань в групах користувачів. Завдяки цьому є можливість реалізувати, за допомогою бота, ефективний метод освіти студента через Телеграм.

У роботі представлений аналіз і розробка чат-бота телеграм для дистанційного навчання студентів. Актуальність роботи також зумовлена умовами самоізоляції через карантин, пов'язаного з пандемією COVID-19 в 2020 році.

У роботі проводиться дослідження методів дистанційного навчання, програмного забезпечення використовується в цих методах, розробка способу навчання за допомогою телеграм-бота, а також розробка самого бота. Бот спрямований на автоматизацію комунікації студента і викладача. При зверненні до боту користувачу буде видаватися інформація, яка потрібна для навчання.

При розробці бота був проведений аналіз предметної області, поставлені завдання, яким повинен відповідати функціонал бота, проведено огляд існуючих аналогів а також наведені приклади використання.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

## 1 ОПИС ОБ'ЄКТА ПРОЄКТУВАННЯ

Творчий задум роботи полягає у відтворенні навчального процесу у дистанційному режимі. Завдяки сучасним можливостям є реальним відтворити навчальний процес, уникаючи безпосереднього контакту студента з викладачем. Розвиток технологій, зокрема бездротового інтернету із високою швидкістю, як 3G, 4G та покриття зв'язком найвіддаленіші точки світу надає ряд переваг, завдяки яким використання методу навчання студентів дистанційно стає більш актуальним. Серед таких переваг ключовими є зберігання навчальних матеріалів та можливість доступу до них цілодобово, а також можливість зв'язку великої кількості користувачів незалежно від місцезнаходження.

Також важливим фактором, що надає актуальності роботі є популяризація таких засобів зв'язку як месенджери – додатки, що використовуються для спілкування на відстані. Наразі месенджери використовуються у багатьох сферах життя: банки використовують для надання допомоги клієнтам, а великі компанії використовують месенджери для організації робочого процесу працівників.

Месенджери доступні для пристроїв, що мають операційну систему, сумісну з додатком. До таких пристроїв відносяться комп'ютери та смартфони, що надає автономності та доступності користувачу. Отже за допомогою месенджера є можливість реалізувати зручний та простий у використанні інструмент для реалізації дистанційної освіти.

У якості програмного забезпечення, за допомогою якого буде проводитись навчання виступає чат-бот. У чат-боті доступний весь функціонал, що має обраний месенджер, а також можливості мови програмування, обраної для розробки. Тож за допомогою бота реалізується автоматизація взаємодії між користувачем та месенджером.

Проєктування будь-якого програмного забезпечення або системи допомагає чітко сформулювати цілі та опрацювати можливості реалізації поставленого завдання. Також це невід'ємна частина для визначення інструментів та найкращих

Зм.	Лист	№ докум.	Підпис	Дата		Лист

рішень для розробки проєкту. Об'єктом проєктування є програмне забезпечення, що має формат чат-бота, яке дозволяє проводити навчання студентів дистанційно.

Метою проєктування є:

- постановка цілей та вимог, яким має відповідати створення програмне забезпечення ;
- визначення переваг та недоліків, що надає обраний формат для реалізації поставленого завдання роботи ;
- аналіз платформ, на базі яких є можливість створення програмного забезпечення та відмінностей обраної платформи від інших ;
- виявлення обмежень, що постають при використанні обраної методики створення необхідних елементів проєкту ;
- аналіз та вибір інструментів, можливостей створення програмного забезпечення.

Щоб розробити бот, необхідно провести декілька етапів проєктування: дослідити вимоги, яким має відповідати бот, і визначити необхідні складові елементи, провести аналіз предметної області та огляд існуючих рішень для визначення необхідного функціоналу.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

## 2 АНАЛІЗ ВИМОГ

Завданням дипломного проекту була розробка бота для дистанційного навчання використовуючи можливості месенджера Telegram. Щоб створити інструмент для повноцінного навчання студентів, по-перше, треба визначити вимоги, яким повинен відповідати цей інструмент та функціонал, який має бути. Щоб це зробити, слід визначити умови, що необхідно створити для отримання освіти студентами.

Основними елементами навчання є:

- вільний доступ до всіх матеріалів, що необхідні для отримання інформації з предмету, а також для виконання практичних робіт (методичок, презентацій, самих завдань) ;
- можливість поводити лекційні та практичні заняття викладача зі студентами ;
- можливість викладачів проводити оцінювання студентів ;
- можливість комунікації викладача із групою студентів, а також із кожним студентом індивідуально ;
- можливість контролю доступу до матеріалів та лекційних, практичних занять.

Також є доцільним розробити додатковий функціонал, що буде виділяти розроблене програмне забезпечення від вже існуючих рішень. Отже з проведеного аналізу можемо скласти лист функцій, які мають бути присутні у боті. Більшість вимог представлена за допомогою обраної платформи для розробки, тобто месенджера Telegram.

За допомогою бота є можливість автоматизувати роботу зі студентами у месенджері. Тобто при використанні бота користувач має отримувати інформацію до предметів та посилання на необхідні ресурси для вивчення матеріалу. Також сповіщати користувача у разі з'явлення нового матеріалу з предмету або

Зм.	Лист	№ докум.	Підпис	Дата		Лист

практичних завдань та термінів, у які необхідно виконати та відправити завдання викладачеві.

Бот має бути зручним та простим у використанні для того, щоб недосвідчені користувачі могли легко почати роботу та отримати всю необхідну інформацію для навчання.

Також важливою вимогою є стабільність роботи, щоб навчальному процесу не заважали помилки у роботі бота або відсутність доступу через погане з'єднання з сервером.

Останньою вимогою є автономність бота. Щоб студент у будь-який час та незважаючи на місцезнаходження зміг отримати усе необхідне для навчання, в нього має бути смартфон або комп'ютер з встановленим Телеграмом.

Проаналізувавши вимоги було виділено основний функціонал, якому має відповідати платформа сам розроблений бот.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

### 3 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 3.1 Переваги Телеграма як платформи

Для чіткого розуміння, слід проаналізувати Telegram та можливості, які надає обрана платформа для реалізації дистанційного навчання. Перевагою використання месенджера Телеграм як платформи для створення бота є готовий графічний інтерфейс, зручність та багатофункціональність, а також підтримка багатьох актуальних та інструментів створення програмного забезпечення. Також месенджер має такі переваги як:

- швидкість відправлення та прийняття повідомлень ;
- приватність – використовуються два види шифрування (E2EE, end-to-end encryption) ;
- відсутність обмежень на розмір повідомлень, а також розмір вкладених файлів ;
- відкритість – Telegram використовує відкритий протокол MTProto і API, отже є вільним до використання, а також розробки свого програмного забезпечення на його базі ;
- має сервери, що розташовані по усьому світі, тому мають високу стабільність і які можна використати для збереження файлів та повідомлень.

Також перевагою є доступний можливість завантаження або видалення файли на сервер, видалення та редагування матеріалів та повідомлень, можливість контролю доступу учасників навчального процесу за допомогою створення різних типів каналів зв'язку, надання спеціальних можливостей користувачам. Для контролю повідомлень та матеріалів, що будуть доступні студентам, використовується функція створення каналів із обмеженими можливостями для користувача. Надсилати матеріали та повідомлення у групу зможуть тільки особи, що мають право адміністратора. Також канали можуть мати майже необмежену кількість учасників. Наприклад, можна створити канал для лекцій на будь-яку

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

кількість студентів з наданням прав адміністратора викладачу, тоді тільки він матиме можливість відсилати повідомлення та необхідні лекційні матеріали.

Telegram не має відео зв'язку, а також групових конференцій, але натомість для проведення лекцій можна використовувати аудіо і відео повідомлення, а також вкладені файли із матеріалом для навчання. Мессенджер не має ліміту на тривалість аудіо повідомлень, а також є зручний пошук у каналах. У такий спосіб доступ до лекційних матеріалів студенти можуть отримати у будь-який час, що є перевагою над лекціями із застосуванням аудіо або відео конференцій. Нещодавно розробники Телеграму анонсували додавання функції відео-конференцій у мессенджер, тому є вірогідність, що незабаром буде можливість реалізації проведення лекцій також у online-режимі. Також є функція створення каналів із обмеженим доступом, що можна використати для того, щоб уникнути втручання сторонніх осіб у навчальний процес.

Найважливіша інформація може бути закріплена завдяки відповідної функції. Закріплене повідомлення буде завжди знаходитись у верхній частині діалогового вікна кожного учасника каналу або групи. Оцінювання студентів буде проводитися або за допомогою вкладених файлів, або використовуючи прямий зв'язок через дзвінки у Telegram. Також дзвінки можна використовувати при необхідності провести усний залік із кожним студентом окремо.

Нещодавно у Telegram з'явилась функція створення опитувань, з обмеженим часом відповіді, що є зручним інструментом для проведення тестів, а також для використання під час практичних занять зі студентами. Таким чином, за допомогою мессенджера є можливість проводити оцінювання студентів під час семестру.

Інформування студентів щодо нових матеріалів з лекцій, оцінок та термінів буде проводити бот, надсилаючи повідомлення персонально кожному студенту.

### 3.2 Телеграм боти

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		



Боти в Телеграм – спеціальні акаунти, створені для автоматичної обробки та відправлення повідомлень. Користувачі взаємодіють з ботами, використовуючи повідомлення, що відправляються у особистому чаті боту або у групових чатах.

Логіка ботів контролюється за допомогою запитів. Користувач відправляє запит у вигляді будь-якого типу повідомлення або команди, далі бот обробляє запит та базуючись на запрограмованих заздалегідь алгоритмах відправляє відповідь.

Функціонал бота визначається користувачем, що програмує бота і обмежується можливостями платформи та обраних інструментів розробки. Деякі приклади використання ботів:

- інтеграція з іншими сервісами. Наприклад, бот може залишати коментарі відправляти вам повідомлення при здійсненні якогось дії або події (Наприклад, GitHub Bot) ;

- утиліти і інструменти. Бот може відображати погоду, перекладати тексти або попереджати про майбутні події ;

- розраховані на одного або багатьох користувачів ігри ;

- соціальні сервіси. Бот може знаходити співрозмовника, ґрунтуючись на спільних інтересах і захопленнях.

Як приклад, GitHub бот створений для інтеграції з репозиторіями на сервісі. При підключенні репозиторія до бота, з'являється можливість лишати коментарі на GitHub за допомогою Телеграму. Також бот періодично перевіряє та інформує користувача про нові події, зв'язані з інтегрованим репозиторієм.

Взаємодія з ботом реалізована за допомогою запитів у форматі команд («/newintegration», «/listintegrations», «/delintegration » та інші). Відповідь на запит відправляється користувачу у формі текстового повідомлення.

На рисунку 3.2.1 зображена взаємодія з GitHub ботом.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

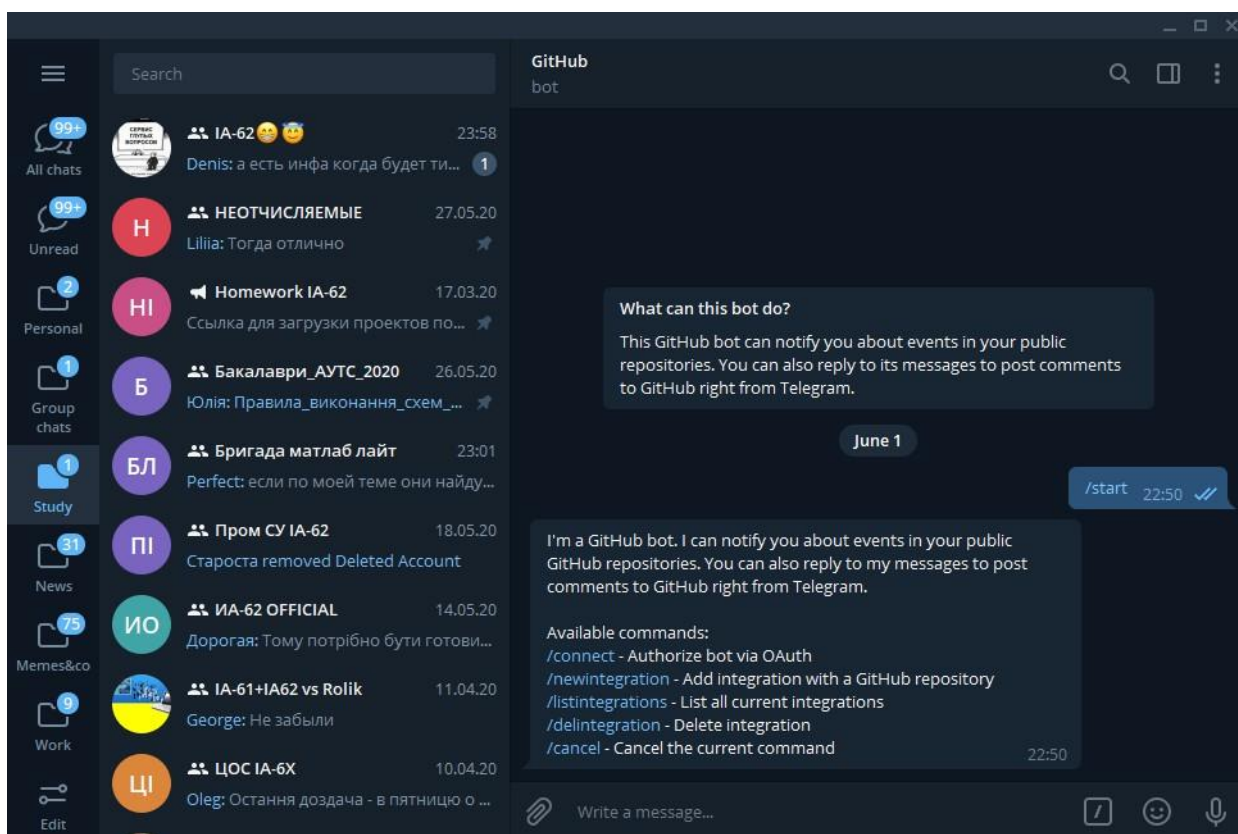


Рисунок 3.2.1 - GitHub бот

### 3.3 Аналіз інструментів створення бота

Для розробки бота використовуються наступні поширені інструменти:

- мови програмування Python, JavaScript або будь-яка, що підтримує відповідну бібліотеку (модуль TelegramBotAPI у Python, Telebot або NodeTelegramBotAPI – у JavaScript) ;
- існуючі боти-конструктори для створення нових ботів та управління створеними як BotFather або ManyBot ;
- конструктори та онлайн платформи для розробки ботів (такі як Puzzlebot, Corezoid).

Для вибору необхідно провести аналіз інструментів. Можливості для створення функціоналу бота дуже обмежені при розробці за допомогою ботів-конструкторів. BotFather необхідно використовувати при реєстрації бота та також

Зм.	Лист	№ докум.	Підпис	Дата	Лист

за допомогою нього виконується управління ботів. За допомогою команд, відправлених BotFather створюються, редагуються та видаляються боти, але список команд не дає можливості розширити функціонал.

Існують також конструктори, наприклад, такий бот, як Manybot. Особистий кабінет сервісу представлений у вигляді бота в Telegram. Конструктор дозволяє створювати меню, підменю, форму зворотного зв'язку, підключати RSS-стрічки і робити розсилки по всім підписаним користувачам. Сервіс ідеально підходить для невеликого проєкту, але створювати меню більше чотирьох рівнів через інтерфейс бота стає складно. Крім цього, бот довго відповідає (або не відповідає зовсім). Оновлення сервісу було зупинено у 2015 році.

Найбільшого функціоналу можна досягти використовуючи для створення бота одну з мов програмування. Весь розроблений функціонал бота відображається на основі інтерфейса Телеграм. Для створення необхідно встановити одну з бібліотек, створених для розробки чат-бота для Телеграма. Тоді за допомогою бота BotFather отримується «токен» (з англійської «token»). Токен – ключ до взаємодії коду та бота, в кожного бота токен унікальний. Для того щоб зміни в коді відображувались на роботі бота треба необхідно пов'язати заздалегідь розроблену логіку модуля з токеном бота. Фактично, токен є авторизацією та реєстрацією бота. Відіславши відповідну команду BotFather, він авторизує юзера, реєструє нового бота у системі та відпраляє токен юзеру («user» - англійською «користувач»). Авторизація проходить за номером телефону користувача, на який він зареєстрований у Telegram.

Конструктори мають більш обмежений функціонал, ніж мови програмування через те, що при розробці конструктор надає тільки базові можливості, а за допомогою мови реалізується весь спектр можливостей API, що надається мові програмування, але конструктори простіші у використанні та мають менший поріг входу. В конструкторах зазвичай використовують діаграми та схеми з можливістю вказувати параметри блокам.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

Одна з найпоширеніших в світі платформа для створення ботів - це Corizoid. Платформа реалізована у формі веб-додатка, ключовою перевагою якої є необмежені можливості інтеграції з зовнішніми сервісами. З його допомогою можна створювати універсальні алгоритми, які працюють у всіх месенджерах відразу. Єдина умова - наявність API. Має зручний інтерфейс та широкий функціонал. Платформа також не безкоштовна, але є тестова версія на 30 днів використання.

З проведеного аналізу Telegram як платформи для проєкту та інструментів, що доступні для використання при створенні бота, виділено переваги, що надаються завдяки функціоналу самій платформи. Також визначені переважні та необхідні інструменти створення. Найбільш доцільно використовувати мову програмування JavaScript через більш широкий спектр можливостей та більшу гнучкість.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

## 4 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

### 4.1 Найпоширеніші платформи для дистанційного навчання

Найбільш актуальні платформи, що використовуються для дистанційного навчання - Zoom, Moodle, Discord. Zoom та Discord реалізовані як додатки для відео та аудіо зв'язку, у Discord також присутні елементи месенджера. Платформа Moodle реалізована як веб-додаток.

### 4.2 Zoom

Zoom пропонує комунікаційне програмне забезпечення, яке об'єднує відеоконференції, онлайн-зустрічі, чат і мобільну кооперативну роботу. Перевагами Zoom є можливість планування навчальних заходів, формат відео зв'язку, а також контроль доступу до конференції.

Недоліками є відсутність можливості завантаження та видалення файлів, низька безпека приватності, нестабільна робота під час високого навантаження, а також платформа не є безкоштовною.

На таблиці 4.2.1 наведений порівняльний аналіз Zoom та Telegram відповідно критеріїв, визначених завдяки аналізу вимог.

Таблиця 4.2.1 – Порівняльний аналіз Zoom та Telegram

Критерій порівняння	Zoom	Telegram
Захист конфіденційності	+	+
Стабільність	-	+
Автономність	-	+

Продовження таблиці 4.2.1

Критерій порівняння	Zoom	Telegram
Можливість планування заходів	+	+
Збереження файлів	-	+
Аудіо зв'язок	+	+
Відео зв'язок та конференції	+	-
Доступність	-	+

З порівняльного аналізу можна зробити висновок, що Telegram більш, ніж Zoom відповідає поставленим задачам завдяки автономності та доступності та наявності функції збереження матеріалів, а також необмеженого доступу до них.

### 4.3 Moodle

Moodle – одне з найпопулярніших у світі середовищ для організації дистанційного навчання. Навчальна платформа створена для того, щоб надати викладачам, адміністраторам та учням інтегровану систему для створення персоналізованих навчальних середовищ.

За допомогою Moodle викладачі можуть створювати та планувати навчальні заходи, опитування, завантажувати та надавати доступ до матеріалів, а також проводити оцінювання студентів.

На рисунку 4.3.1 зображений інтерфейс особистого кабінету користувача Moodle.

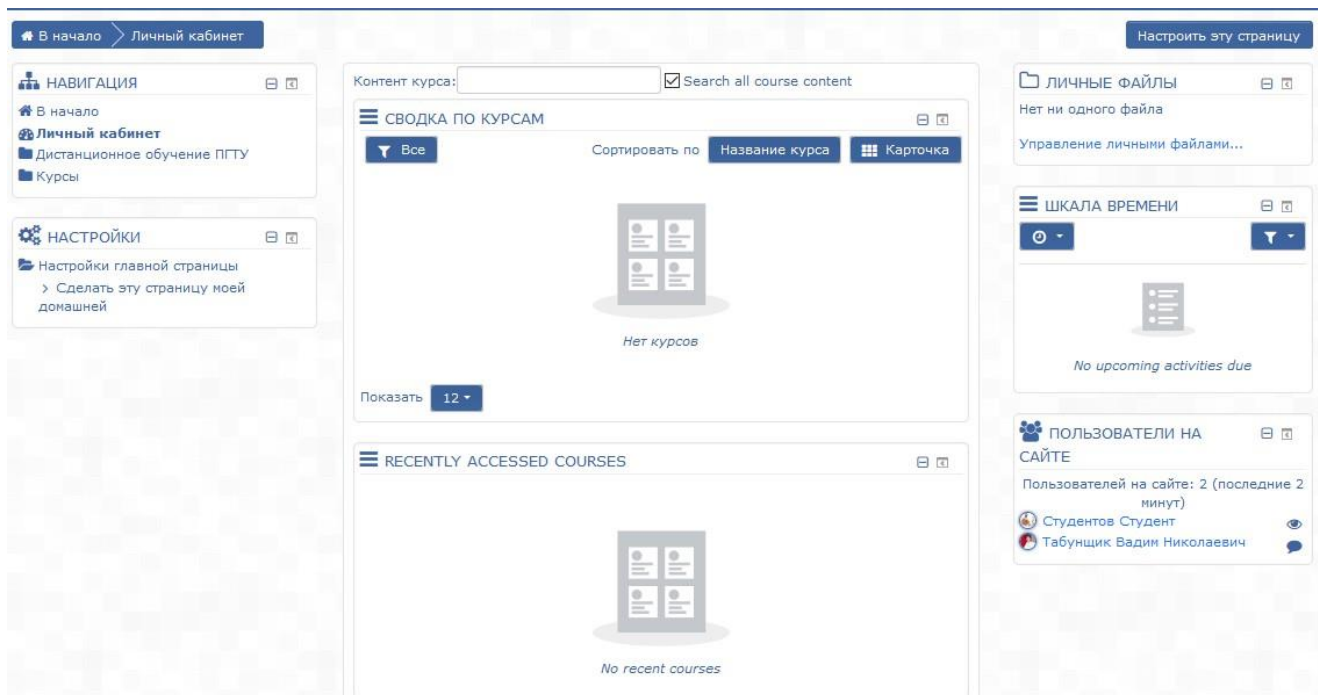


Рисунок 4.3.1 – Особистий кабінет користувача Moodle

Недоліками платформи є неможливість зв'язку між студентом та викладачем, а також незручність у використанні. Для адміністрування Moodle треба мати навички веб-розробки, що робить його більш складним у використанні і менш доступним для недосвідчених користувачів. Для використання Moodle, платформу необхідно завантажити на написаний заздалегідь сервер. Також не має технічної підтримки.

На таблиці 4.3.1 наведено порівняльний аналіз Moodle з Telegram відповідно до необхідних вимог

Таблиця 4.3.1 – Порівняльний аналіз Moodle та Telegram

Критерій порівняння	Moodle	Telegram
Захист конфіденційності	+	+
Стабільність	+	+
Автономність	+	+

Продовження таблиці 4.3.1

Критерій порівняння	Moodle	Telegram
Можливість планування заходів	+	+
Збереження файлів	-	+
Аудіо зв'язок	-	+
Відео зв'язок та конференції	-	-
Доступність	-	+

Із порівняльного аналізу платформ виявлені переваги використання Telegram відповідно до вимог проєкта, такі як: доступність через простоту у використанні, наявність можливості комунікації студентів та викладача, а також обміну файлами та збереження їх у сховищі на сервері.

#### 4.4 Discord

Discord – додаток-месенджер з підтримкою VoIP (зв'язок за допомогою використання протокола IP) і відеоконференцій. Додаток здатний організовувати голосові конференції з налаштуванням каналу зв'язку і працювати за принципом push-to-talk, створювати публічні і приватні чати для обміну текстовими повідомленнями.

Перевагами Discord є можливість аудіо та відео зв'язку між студентом та викладачем, висока швидкість та можливість завантаження файлів та повідомлень.

На рисунку 4.4.1 зображений інтерфейс Discord.



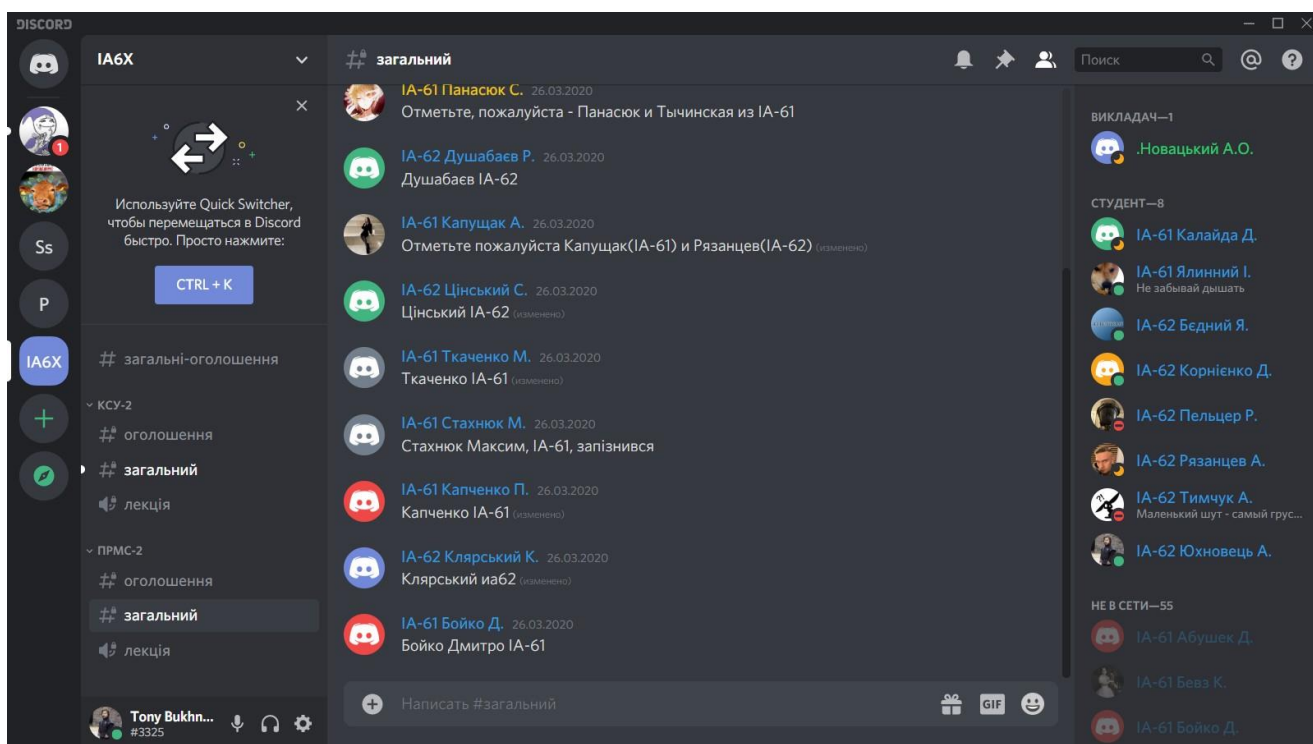


Рисунок 4.4.1 – Інтерфейс Discord

Недоліком є неможливість планування заходів, збереження дзвінків, а також у додатку немає достатнього контролю доступу до робочих груп.

На таблиці 4.4.1 представлений порівняльний аналіз Telegram та Discord.

Таблиця 4.4.1 – Порівняльний аналіз Telegram та Discord

Критерій порівняння	Discord	Telegram
Захист конфіденційності	-	+
Стабільність	+	+
Автономність	+	+
Збереження файлів	+	+
Відео зв'язок та конференції	+	-

#### Продовження таблиці 4.4.1

Критерій порівняння	Discord	Telegram
Аудіо зв'язок	+	+
Доступність	+	+
Можливість планування заходів	-	+

У проведеному порівняльному аналізі виявлені недоліки та переваги у виборі основної платформи розробки бота. Telegram має перевагу у захисті конфіденційності та можливості планування заходів. Головна перевага Discord – наявність відео та аудіо конференцій, але у додатку не представлена можливість записувати аудіо повідомлень, а також відсутня можливість збереження дзвінків, тому за допомогою аудіо повідомлень, що присутні у Telegram, є можливим досягти мети проєкта, уникаючи використання конференцій.

#### 4.5 Висновки до розділу

Проведений огляд існуючих рішень дав представлення методів реалізації поставленої задачі проєкту, а також виявив основні переваги використання цих методів. Також був проведений порівняльний аналіз обраної платформи з аналогами та можливостей, наданих для реалізації дистанційного навчання. Було зроблено висновок, що Telegram найкраще відповідає цілям та вимогам проєкту завдяки більшій доступності, наявності необхідного функціоналу.

## 5 РОЗРОБКА TELEGRAM-БОТА

### 5.1 Реєстрація чат-бота та отримання токена

Для реєстрації бота необхідно провести авторизацію користувача у спеціальному боті. Для цього необхідно знайти його за унікальним ім'ям @BotFather та розпочати авторизацію за допомогою команди «/start». Бот авторизує користувача за номером телефону. Також бот активізується для роботи з користувачем та видає лист доступних команд.

У таблиці 5.1.1 наведений лист доступних команд BotFather.

Таблиця 5.1.1 – Лист команд BotFather

Команда	Опис
/newbot	Створення нового бота
/mybots	Внесення змін у вже існуючих ботів
/setname	Присвоєння імені
/setabouttext	Зміна опису бота
/setuserpic	Присвоєння картинки
/setcommands	Зміна листа команд
/deletebot	Видалення бота
/token	Генерація токена авторизації
/revoke	Відкликати токен доступу бота
/setline	Режим вбудованих запитів
/setlinegeo	Режим вбудованої геолокації

## Продовження таблиці 5.1.1

Команда	Опис
/setlinefeedback	Налаштування зворотнього зв'язку
/setjoininggroups	Налаштування можливості додавання бота в групи
/setprivacy	Налаштування приватного режиму
/mygames	Лист ігор
/listgames	Створення нової гри
/editgame	Зміна існуючої гри
/deletgame	Видалення гри

Далі проводиться реєстрація нового бота, відправивши команду «/newbot» та вибір імені бота. Ім'я має включати в кінці приставку «\_bot». Після вибору імені отримуємо токен. Далі за допомогою BotFather виконується управління ботом. За допомогою спеціальних команд встановлюється рисунок як фотографія профіля бота, також встановлюється лист активних команд, ім'я та загальна інформація про бота.

На виході отримали унікальний токен бота для роботи з HTTP API – «1154060366:AAGa3JUXruJbFx ZZ3pUbLparmTDH2A8M9E».

На рисунку 5.1.1 зображений результат взаємодії з FatherBot та виконання команди «/newbot».

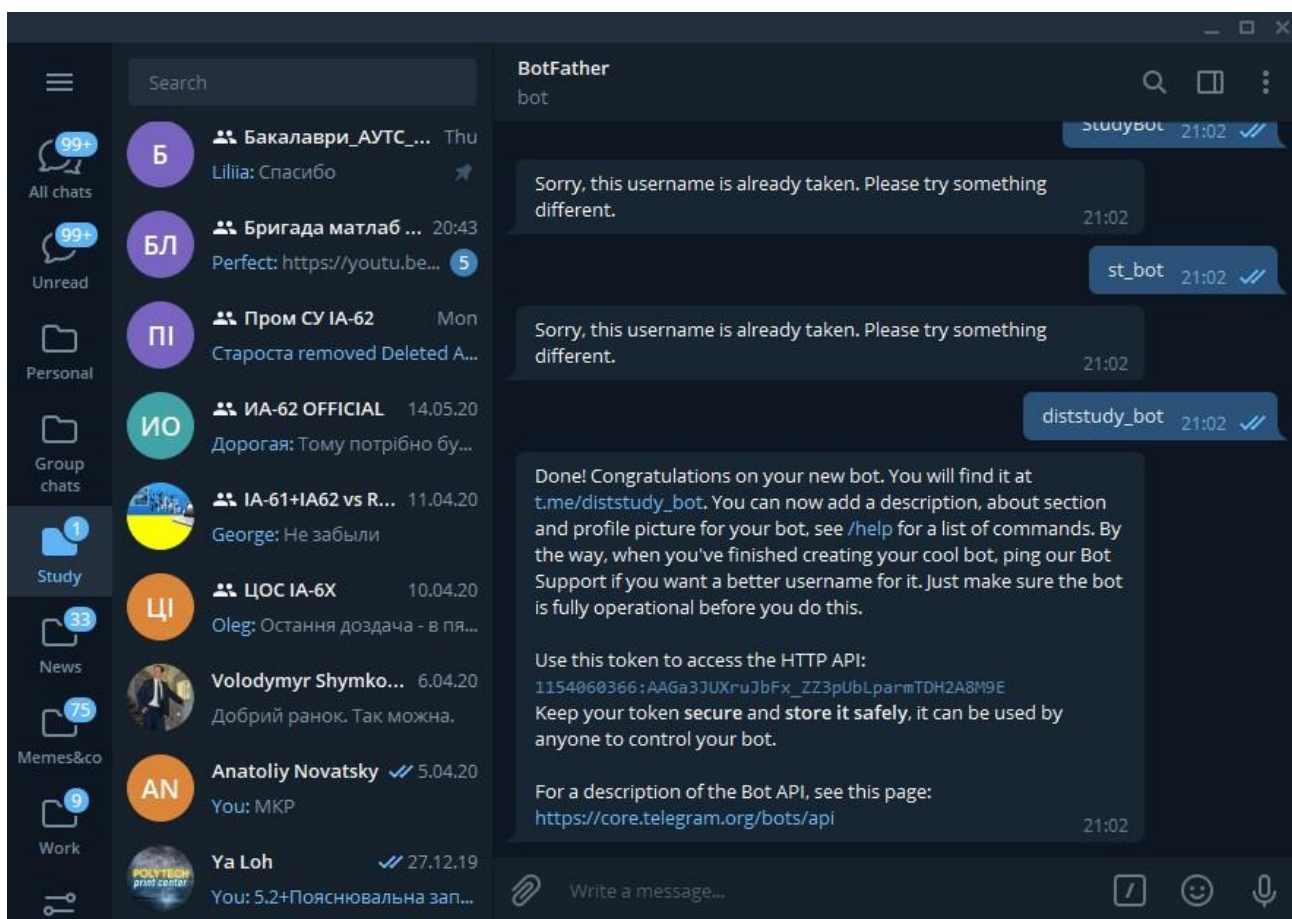


Рисунок 5.1.1 – Результат виконання команди «/newbot»

## 5.2 Середовище розробки

В якості середовища розробки була обрана IDE Visual Studio Code через підтримку необхідних мов програмування (Python, Node.js) та високі можливості написання програмного забезпечення. Visual Studio Code представлена у форматі редактора коду для крос-платформної розробки веб-додатків і хмарних тек, що сумісна з Windows, MacOS та Linux.

IDE має такі переваги:

- вбудований відладчик, що дозволяє знаходити та виправляти помилки в коді ;
- можливість працювати з репозиторіями системи контролю версій Git, що є одною з ключових переваг через надання можливості повернутися, при

Зм.	Лист	№ докум.	Підпис	Дата		Лист

необхідності, до потрібної версії коду, а також, наприклад, при завантаженні додатку на сервер ;

- автодоповнення, що спрощує написання та редагування коду при написанні програмного забезпечення ;
- кросс-платформенність, що дозволяє використовувати Visual Studio Code програмуючи на декількох мовах програмування та для різних проєктів ;
- безкоштовність та доступність у використанні.

З аналізу середовища розробки зроблено висновок, що завдяки ряду переваг Visual Studio Code є доцільним використання IDE для створення бота.

### 5.3 Модуль Telebot

Для написання боту необхідно інтегрувати одну з бібліотек Node.js, що використовуються для взаємодії з офіційним API Telegram Bot. На вибір доступні для використання наступні модулі:

- Telegraf ;
- Node-Telegram-bot ;
- Tgapi ;
- Telebot ;
- Botgram ;
- Slimbot .

Проаналізувавши модулі виявлено, що деякі з них не надають достатнього функціоналу розробки. Наприклад, модулі Botgram та Node-Telegram-bot направлені на створення ботів та взаємодію з ним, але не мають команд для підключення баз даних, отримання даних користувача, а також багато інших функцій, що необхідні для виконання завдання.

Найбільш доцільним для розробки потрібного бота є Telebot через високу гнучкість, достатній функціонал, а також обширний спектр методів і функцій.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

### 5.3.1 Події TeleBot

Для обробки всіх можливих подій TeleBot використовується метод «on (<event>, <function>)» [4-5].

В таблиці 5.3.1 наведений лист подій TeleBot.

Таблиця 5.3.1.1 – Події TeleBot

Подія	Опис
/*	Будь-яка команда користувача
/<cmd>	Спеціальна команда
start	Бот розпочав роботу
stop	Бот припинив роботу
reconnecting	Повторне підключення бота
reconnected	Повторне підключення успішне
update	Оновлення
tick	Шаг
error	Виявлена помилка
inlineQuery	Дані вхідного запиту
chosenInlineResult	Результат обраного вхідного запиту

## Продовження таблиці 5.3.1.1

Подія	Опис
callbackQuery	Дані виклику кнопки
shippingQuery	Доставка запиту
preShippingQuery	Перевірка вхідного запиту

### 5.3.2 Модифікатори TeleBot

Для обробки даних, перш ніж передавати їх у подію, використовуються модифікатори за допомогою метода «mod(<name>, <fn>)» [4-5].

У таблиці 5.3.2.1 наведений лист модифікаторів TeleBot.

Таблиця 5.3.2.1 – лист модифікаторів TeleBot

Модифікатор	Опис
property	Властивості
updateList	Список оновлень за шаг
message	Обробка будь-якого типу повідомлення
<type>	Конкретний тип повідомлення
update	Кожне оновлення



### 5.3.3 Методи TeleBot

Реалізація основного функціоналу при розробці бота будується на основі методів, що доступні в бібліотеці. TeleBot має великий спектр методів для використання.

У таблиці 5.3.3.1 приводиться лист методів TeleBot, що використовувались при розробці бота.

Таблиця 5.3.3.1 – Лист використаних методів TeleBot

Метод	Опис
on()	Обробляє події
answerList()	Створює об'єкт списку відповідей вбудованого запиту
inlineQueryKeyboard()	Створює об'єкт клавіатури для листу відповідей
keyboard()	Створює об'єкт розмітки клавіатури
start()	Починає завантаження оновлень
button()	Додає кнопку
inlineButton()	Створює об'єкт кнопки
event()	Викликає обробник подій

### Продовження таблиці 5.3.3.1

Метод	Опис
Stop()	Припиняє завантаження оновлень
inlineKeyboard()	Створює вбудований об'єкт клавіатури для повідомлень
mod()	Додає модифікатор
modRun()	Приводє до дії модифікатор

## 5.4 Розробка веб-серверу

### 5.4.1 HTTP-запити

HTTP - це клієнт-серверний протокол, що дозволяє отримувати різні ресурси. Запити відправляються одною стороною - учасником обміну. Кожен запит відправляється на сервер, який обробляє його і повертає відповідь. Між запитом і відповіддю, як правило, існують численні посередники, звані проксі, які виконують різні операції і працюють як шлюзи або кеш (з англійської - «cache»). Найчастіше в якості учасника виступає веб-браузер, але сфера використання також розповсюджується і на ботів.

На рисунку 5.4.1.1 зображений шлях, що проходить HTTP-запит після відправлення.

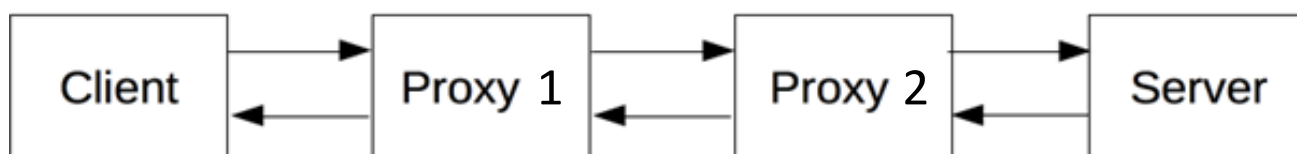


Рисунок 5.4.1.1 – Шлях HTTP-запиту [7]

Зазвичай між браузером і сервером набагато більше різних пристроїв-посередників, які відіграють якусь роль в обробці запиту: маршрутизатори, модеми і так далі. Завдяки тому, що мережа побудована на основі системи рівнів взаємодії, ці посередники "заховані" на мережевому і транспортному рівнях. У цій системі рівнів HTTP займає верхній рівень, який називається "прикладним" (або "рівнем додатків").

Запити містять такі елементи:

- HTTP-метод, зазвичай дієслово подібно GET, POST або іменник, як OPTIONS або HEAD, що визначає операцію, яку клієнт хоче виконати ;
- шлях до ресурса ;
- версію HTTP-протоколу ;
- заголовки, що представляють додаткову інформацію для сервера ;
- тіло, для деяких методів, таких як POST, яке містить відправлений ресурс.

Відповіді містять наступні елементи:

- версію HTTP-протоколу ;
- HTTP код стану - повідомляє про успішність запиту або причини невдачі ;
- повідомлення стану - короткий опис коду стану ;
- HTTP заголовки, подібно заголовкам в запитах ;
- необов'язково: тіло, що містить ресурс, що пересилається.

Для створення серверу на Node.js, мова програмування взаємодіє з вбудованим модулем HTTP. Але HTTP вкрай низькорівневий: створення складного веб-додатку з використанням тільки цього модуля коду потребує занадто багато роботи.

#### 5.4.2 Express.js

Для спрощення роботи над сервером використовуються фреймворки (з англійської – «framework»). Є безліч фреймворків, ось найпопулярніші:

- express ;

Зм.	Лист	№ докум.	Підпис	Дата		Лист

- hapi ;
- koa ;
- restify.

Для розробки серверу для бота використаємо Express, так як саме для нього існує безліч модулів в NPM [14].

Express - веб-фреймворк Node.js, що лежить в основі багатьох інших популярних фреймворків. Він забезпечує механізми:

- написання обробники для запитів з різними HTTP-запитами за різними URL-адресами (маршрутами) ;
- інтеграція з механізмами рендеринга «view» для генерації відповідей шляхом вставки даних в шаблони ;
- завдання загальних параметрів веб-додатку, такі як порт, який використовується для підключення, і розташування шаблонів, які використовуються для візуалізації відповіді ;
- додавання додаткової обробки запитів "middleware" в будь-якій точці конвеєра обробки запитів.

Middleware - проміжний обробник. Запит проходить через три проміжних обробника. Кожен обробник може змінити цей запит, а потім, ґрунтуючись бізнес-логікою, третій middleware відправить відповідь, або запит потрапить в обробник відповідного роут-хендлера ( з англійської «route-handler» - обробник маршруту)

На Рисунку 5.4.2.1 ви можете побачити, як запит йде через умовний Express-додаток.



Рисунок 5.4.2.1 – Шлях запиту з middleware [13]

На рисунку 5.4.2.2 представлена реалізація middleware в коді.

```

1  const express = require('express')
2  const app = express()app.use((request, response, next) => {
3      console.log(request.headers)
4      next()
5  })app.use((request, response, next) => {
6      request.chance = Math.random()
7      next()
8  })app.get('/', (request, response) => {
9      response.json({
10         chance: request.chance
11     })
12 })app.listen(3000)

```

Рисунок 5.4.2.2 - Реалізація middleware в коді

Метод «`app.use()`» - описання middleware. Цей метод приймає функцію з трьома параметрами, перший з яких є запитом, другий - відповіддю, а третій - функцією зворотнього виклику «`next`». Виклик «`next`» сигналізує Express про те, що він може переходити до наступного проміжного обробника ;.

Перший проміжний обробник тільки проводить логювання і миттєво викликає наступний .

Другий додає додаткову властивість до запиту - це одна з найбільш потужних функцій шаблону middleware. Проміжні обробники можуть додавати додаткові дані до об'єкта запиту, який можуть зчитувати та змінювати middleware, розташовані нижче.

Як і у всіх фреймворках, в Express правильна обробка помилок має вирішальне значення. Необхідно створити спеціальний проміжний обробник - middleware з чотирма вхідними параметрами.

Оновлення помилок повинно бути останньою функцією, доданою з допомогою методу «`app.use()`».

Наступний обробник помилки виконує коллбек (з англійської - «callback»). Він може використовуватись для об'єднання декількох обробників помилок.

На рисунку 5.4.2.3 зображена обробка помилок middleware.

```

1  const express = require('express')
2  ✓ const app = express()app.get('/', (request, response) => {
3    throw new Error('...')
4  ✓ })app.use((err, request, response, next) => {
5    // логювання помилки console.log
6    console.log(err)
7    response.status(500).send('Щось зламалось')
8  })app.listen(3000)
9  |

```

Рисунок 5.4.2.3 – Обробка помилок middleware

Хоча сам Express досить невеликий за обсягом, розробники створили сумісні пакети проміжного програмного забезпечення для вирішення практично будь-якої проблеми веб-розробки. Існують бібліотеки для роботи з файлами cookie, сесіями, логінами користувачів, параметрами URL, даними POST, заголовками безпеки і багатьом іншим.

Методи маршруту походять від одного з методів HTTP і використовуються з екземпляром класу «express». Це методи «get()», «post()» і «all()».

У таблиці 5.4.2.1 приведені методи об'єкту «res», що можуть надіслати відповідь клієнту та припинити цикл запит-відповідь.

Таблиця 5.4.2.1 – Методи об'єкту «res»

Метод	Опис
res.download()	Завантаження файлу
res.end()	Закінчити процес відповіді
res.json()	Надіслати відповідь JSON.
res.jsonp()	Надіслати відповідь JSON з підтримкою JSONP
res.redirect()	Переадресація запиту



груп, таблиця предметів, що відповідають групі, і таблиця навчальних матеріалів, що буде відправляти бот.

У таблиці 5.5.1.1 наведені елементи таблиці студентів.

Таблиця 5.5.1.1 – Елементи таблиці «Студенти»

Елемент	Опис
Студент	Ім'я користувача або «нік»
Id студента	Унікальний номер студента
Id групи	Унікальний номер групи, до якого прив'язан студент

У таблиці 5.5.1.2 наведені елементи таблиці груп.

Таблиця 5.5.1.2 – Елементи таблиці «Група»

Елемент	Опис
Id групи	Унікальний номер групи для ідентифікації
Група	Назва групи
Чат групи	Посилання на чат для спілкування між групами

У таблиці 5.5.1.3 наведені елементи таблиці предметів.

Таблиця 5.5.1.3 – Елемент таблиці «Предмети»



Елемент	Опис
Предмет	Назва предмету
Id предмету	Унікальний номер предмету
Лекційні канали предмету	Посилання на канали з лекціями предмета
Практичні чати предмету	Посилання на чати для практичних занять з предмету

У таблиці 5.5.1.4 наведені елементи таблиці викладачів.

Таблиця 5.5.1.4 – Елемент таблиці «Викладачі»

Елемент	Опис
Викладач	Ім'я викладача
Id викладача	Унікальний ідентифікаційний номер викладача
Id предмета	Унікальний номер предмета або предметів, що прив'язані до викладача
Контакт викладача	Посилання на профіль або інший спосіб зв'язку з викладачем

У таблиці 5.5.1.5 наведені елементи таблиці навчальних матеріалів.

Таблиця 5.5.1.5 – Елементи таблиці «Матеріали»

Зм.	Лист	№ докум.	Підпис	Дата		Лист

Елемент	Опис
Матеріал	Матеріал для відправлення
Id матеріалу	Унікальний номер матеріалу
Id предмету	Унікальний номер предмету
Дедлайн	Строки виконання практичних або лабораторних робіт

Коли користувач починає роботу з ботом, він має вказати свої дані (ім'я, прізвище, навчальна група). Далі проводиться пошук по базі за назвою групи і реєстрація користувача, а також закріплення введеної групи за студентом, якщо група існує. Якщо група відсутня в базі, бот повідомить користувача про невірно вказані дані.

Після ідентифікації групи бот витягує з таблиці, за допомогою унікального номера групи, лист предметів, що відповідають групі, а також лист викладачів, що ведуть предмети.

Якщо предмет прив'язаний до декількох груп, поле «id групи» може бути представлено у вигляді масиву унікальних ідентифікаторів групи. До кожного предмету прив'язаний лист навчальних матеріалів з предмету за допомогою id предмета або масиву за декількох id.

Наступним кроком після отримання листа предметів для групи є витяг списку навчальних матеріалів, доступних з кожного предмету та викладачів які закріплені за допомогою ідентифікаційного коду за предметом. Далі бот відправляє користувачу лист предметів та відповідних до них матеріалів і викладачів.

Повна схема зв'язків бази даних представлена у додатку В.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

### 5.5.2 MySQL та SQL

Для створення та роботи з базами даних використаємо один з найпоширеніших систем управління базами даних – MySQL.

MySQL це система керування базами даних з відкритим вихідним кодом (СУРБД) з моделлю «клієнт-сервер». СУРБД - це програмне забезпечення або служба, яка використовується для створення та управління базами даних на основі реляційної моделі. Комп'ютери, які встановлюють і запускають програмне забезпечення СУРБД, називаються клієнтами. Коли їм потрібно отримати доступ до даних, вони підключаються до сервера СУРБД. Це система «клієнт-сервер» [15].

В якості основного мови для взаємодії з базою даних використовується SQL. SQL – декларативна мова програмування, призначена для збереження, маніпуляції і отримання даних, які зберігаються в реляційній базі даних. При виконанні команди SQL, система визначає найкращий спосіб для обробки запиту, і модуль управління («двигун») SQL обчислює, як інтерпретувати завдання. У цей процес входять наступні компоненти:

- запит (Query) ;
- диспетчер (Dispatcher) ;
- двигуни оптимізації (Optimization Engines) ;
- двигун запитів (Classic Query Engine) ;
- двигун SQL-запитів (SQL Query Engine).

Двигун запитів оброблює всі не-SQL-запити, а двигун SQL-запитів не оброблює логічні файли.

Створення необхідної для бота бази даних відбувається із використанням команд SQL. Нижче приведений лист команд, що використані у проєкті.

Стандартні команди SQL поділяються на три категорії: для визначення (Data Definition Language або DLL), для управління (DML – Data Manipulation Language), та контролю (DCL – Data Control Language).

Зм.	Лист	№ докум.	Підпис	Дата		Лист

DLL команди використовуються для створення, видалення або зміни об'єктів баз даних.

У таблиці 5.5.2.1 приведений лист команд DLL

### Таблица 5.5.2.1 – DLL команди

Команда	Опис
CREATE	Створює нову таблицю, вид таблиці або інший об'єкт в БД
ALTER	Змінює існуючий об'єкт БД (наприклад, таблицю)
DROP	Видаляє зазначену таблицю, вид таблиці або інший об'єкт БД

DML команди використовуються для управління даними в таблицях або інших об'єктах бази даних.

У таблиці 5.5.2.1 приведений лист команд DML

### Таблица 5.5.2.1 – DML команди

Команда	Опис
SELECT	Отримує певні дані з однієї або декількох таблиць
INSERT	Створює запис
UPDATE	Змінює запис

## Продовження таблиці 5.5.2.1

Команда	Опис
DELETE	Видаляе запис

DCL команди використовуються для надання або відміни привілеїв користувача.

У таблиці 5.5.2.1 приведений лист команд DCL

### Таблица 5.5.2.1 – DCL команди

Команда	Опис
GRANT	Надає усі привілеї користувачу
REVOKE	Відмінляє привілеї, надані користувачу

Використовуючи приведені SQL команди та СУБД MySQL створюємо базу даних для бота зі студентів, викладачів, груп, предметів та навчальних матеріалів.

## 5.6 Telegram Bot API

### 5.6.1 Авторизація бота

Окрім модулю TeleBot, при розробці неможливо обійтися без використання стандартного Telegram Bot API модуля. Першим кроком необхідно зв'язати код з

ботом. Для цього використаємо токен бота. Створимо файли «config.json» та «index.js». У config.json пропишемо токен, а в index.js створимо константу «token» та присвоїмо їй дані з config.json, далі створимо константу «bot» вказавши в параметрах вже отриманий токен. Також необхідно прописати «polling=on». Polling необхідний щоб бот намагався не припиняти роботу при виникненні будь-яких помилок.

### 5.6.2 Запити

Telegram Bot API підтримує GET та POST HTTP-запити. Відповідь на запит містить об'єкт JSON, у якому завжди є поле «ok» типа boolean і може мати необов'язковий рядок «description» з описом результату. Якщо «ok» дорівнює true, запит був успішним, і результат запиту можна знайти в полі «result». У разі невдалого запиту "ok" дорівнює false, а помилка пояснюється в описі. Повертається також поле «error\_code» типу integer, але його вміст може змінитися в майбутньому. Деякі помилки можуть також мати необов'язкове поле «params» типу «ResponseParameters», що може допомогти автоматично впоратися з помилкою. Запити повинні бути виконані за допомогою кодування UTF-8. Усі об'єкти у Telegram Bot API представлені у форматі об'єкта JSON.

Об'єкт - це неупорядкований набір імен або значень. Об'єкт JSON починається з лівої фігурної дужки і закінчується правою. Кожне ім'я супроводжується двокрапкою та пари імен або значень, розділені комою.

На рисунку 5.6.2.1 представлене схематичне зображення об'єкта JSON.

Зм.	Лист	№ докум.	Підпис	Дата		Лист

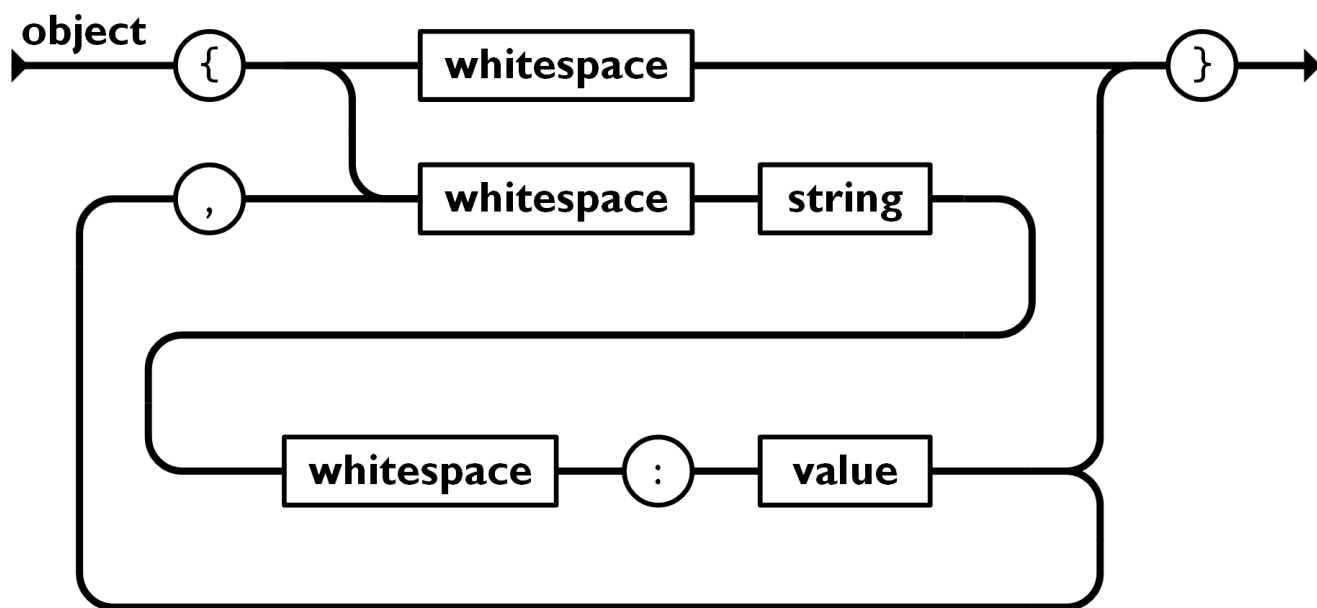


Рисунок 5.6.2.1 - Схематичне зображення об'єкта JSON [3]

### 5.6.3 Об'єкти

В Telegram Bot API [1] безліч функцій, типів та об'єктів, але при розробці бота, з необхідним відповідно вимогам функціоналом, досить використати деякі з них. Нижче приводиться лист об'єктів, що задіяні у проєкті.

Об'єкт «User» представляє користувача або бота Telegram.

У таблиці 5.6.3.1 наведені поля об'єкта User.

Таблиця 5.6.3.1 – Поля об'єкту User

Поле	Тип	Опис
id	Integer	Унікальний ідентифікатор

Продовження таблиці 5.6.3.1

Поле	Тип	Опис
is_bot	Boolean	True, якщо бот
first_name	String	Ім'я користувача
last_name	String	Фамілія користувача
username	String	«Нік» користувача

Об'єкт «Chat» представляє групи та канали у Telegram.

У таблиці 5.6.3.2 наведені поля об'єкта Chat.

Таблиця 5.6.3.2 – Поля об'єкту Chat

Поле	Тип	Опис
id	Integer	Унікальний ідентифікатор чату
type	String	Тип чату (група, приватний чат, канал)
title	String	Назва
username	String	«Нік» групи
last_name	String	Друге ім'я, якщо існує



Продовження таблиці 5.6.3.2

Поле	Тип	Опис
first_name	String	Ім'я групи

Об'єкт «Message» представляє повідомлення.

У таблиці 5.6.3.3 наведені поля об'єкту Message.

Таблиця 5.6.3.3 – Поля об'єкту Message

Поле	Тип	Опис
message_id	Integer	Ідентифікатор повідомлення
from	User	Відправник
date	Integer	Дата повідомлення
contact	Contact	Повідомлення із контактом
document	Document	Повідомлення з документом
text	String	Текстове повідомлення

Об'єкт «Document» представляє файли.

У таблиці 5.6.3.4 наведені поля об'єкту Document.

Таблиця 5.6.3.4 – Поля об'єкту Document

Поле	Тип	Опис
file_id	String	Ідентифікатор файлу
file_unique_id	String	Ідентифікатор файлу, що має бути однаковим у часі та для різних ботів.
thumb	PhotoSize	Ескіз документа, заданий відправником
file_name	String	Ім'я файлу, задане відправником
mime_type	String	MIME-тип файлу, заданий відправником
file_size	Integer	Розмір файлу

Об'єкт «File» представляє файли, що готові до завантаження.

У таблиці 5.6.3.5 наведені поля об'єкту File.

### Таблиця 5.6.3.5 – Поля об'єкту File

Поле	Тип	Опис
file_id	String	Ідентифікатор
file_unique_id	String	Ідентифікатор файлу, що має бути однаковим у часі та для різних ботів.

file_path	String	Місцезнаходження файлу (шлях до файлу)
file_size	Integer	Розмір файлу

Об'єкт «KeyboardButton» представляє кнопки.

У таблиці 5.6.3.6 наведені поля об'єкту KeyboardButton

Таблиця 5.6.3.6 – Поля об'єкту KeyboardButton

Поле	Тип	Опис
text	String	Текст на кнопці
request_contact	Boolean	Запит номера користувача
request_location	Boolean	Запит локації користувача

Об'єкт «BotCommand» представляє команди, доступні боту.

У таблиці 5.6.3.7 наведені поля об'єкту BotCommand.

Таблиця 5.6.3.7 – Поля об'єкту BotCommand

Поле	Тип	Опис
command	String	Текст команди

### Продовження таблиці 5.6.3.7

Поле	Тип	Опис
description	String	Опис команди

### 5.6.4 Методи

Окрім POST та GET методів, Telegram Bot API присутній ряд методів для реалізації можливостей бота. Усі методи в API не залежать від регістру. Нижче наведені методи, що використанні при створенні бота.

Метод «getChatMember()» використовується для отримання інформації щодо користувача в чаті. Метод має два параметри: «chat\_id», що може бути типу Integer або String і є ідентифікатором чату і передається у метод для визначення потрібного діалогу ботом та метод «user\_id» типу Integer, що є унікальним ідентифікатором користувача.

Для зміни та виводу листа команд, що виконує бот використовуються методи «setMyCommands()» та «getMyCommands()» відповідно. Метод setMyCommands() має параметр «commands» типу масив і є JSON-серіалізованим списком команд бота, які можна встановити.

Метод «getFile()» використовується для отримання інформації про файл та підготувати його до завантаження. Має параметр «file\_id» типу string, що є унікальним ідентифікатором файлу.

Метод «getMe()» дозволяє протестувати бот. Повертає інформацію про бота в формі об'єкта User.

Метод «sendMessage()» використовується для відправлення текстових повідомлень. Метод має наступні параметри:

- «chat\_id» типу Integer або String, що є ідентифікатором чату;
- «text» типу String, що є текстом повідомлення, яке передає бот за допомогою методу;
- «parse\_mode» типу String є режимом синтаксичного аналізу, що використовується до тексту повідомлення;
- «reply\_to\_message\_id» типу Integer використовується для знаходження ідентифікатору повідомлення, на яке була відповідь;
- «disable\_web\_page\_preview» та «disable\_notification» типу Boolean, для відключення попереднього огляду посилань та сповіщення про повідомлення відповідно;
- «reply\_markup», що може бути типів InlineKeyboardMarkup, ReplyKeyboardMarkup, ReplyKeyboardRemove або ForceReply, що використовується для налаштування додаткових параметрів інтерфейсу із використанням режиму вбудованої клавіатури або «Inline Mode».

Метод «sendDocument()» використовується для відправлення файлів, що є одною з ключових функцій бота. Для використання методу необхідно вказати деякі необхідні параметри, а також є можливість передати додаткові при необхідності.

Лист параметрів методу:

- «chat\_id» типу Integer або String, що є ідентифікатором чату ;
- «document» типу InputFile або String, за допомогою цього параметру у метод передається файл, що необхідно відправити, тип залежить від способу передачі файлу в метод ;
- «thumb» типу InputFile або String, за допомогою що є ескізом надісланого файлу ;
- «caption» типу String, за допомогою якого вказується підпис до документа, що надсилається ;
- «parse\_mode» типу String, що є режимом синтаксичного аналізу тексту ;

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

- а також «reply\_to\_message\_id», «disable\_web\_page\_preview», «disable\_notification» та «reply\_markup» аналогічні параметрам повідомлення.

Метод «sendPhoto()» використовується для відправлення фото матеріалів та рисунків. Має аналогічні параметри, що й sendDocument(), за виключенням «thumb» та «document», замість документу передається параметр «photo», який аналогічно може бути типу InputFile або String. Необхідно зазначити, що відправляти фото можливо також і у якості документа, але відправлення через такий спосіб ускладнює пошук у матеріалах чату.

Останній метод Telegram Bot API, що був використаний при розробці бота – «sendVideo()», що дозволяє відправляти відео файли. Відмінно від фото має додаткові параметри:

- «duration» типу Integer, що передає у метод тривалість відео ;
- «width» типу Integer, що вказує на ширину відео ;
- «height» типу Integer, що є висотою відео.

Також аналогічно фото та документу має обов'язкові параметри «chat\_id» та «video», що вказують відповідно ідентифікатор чату і відео шлях чи посилання до місцезнаходження відео файлу.

Як і «sendDocument()» має параметри thumb і caption, що є ескіз та підпис відео, а також «reply\_to\_message\_id», «disable\_web\_page\_preview», «disable\_notification» та «reply\_markup».

Використання методів представлено у програмному коді у додатку А.

## 5.7 Heroku сервер

Щоб бот був активний весь час необхідно завантажити його на сервер. Існує безліч безкоштовних хмарних рішень, з яких виберемо Heroku через безкоштовність сервісу, достатнього функціоналу для реалізації задуму та технічної підтримки.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

Heroku – одна з перших хмарних платформ. Влаштована на моделі PaaS (Platform as a Service або «платформа як послуга»). PaaS - модель надання хмарних обчислень, при якій споживач отримує доступ до використання ресурсів та використання платформ. Усі ресурси та платформи контролює провайдер, а також надає доступ користувачу до ряду функціоналу [9].

У першу чергу необхідно створити два файли web.js та package.json. Створити їх необхідно у головному каталозі бота.

На рисунку 5.7.1 зображен код, що містить web.js.

```
1 var express = require('express');
2 var app = express();
3 var path = require('path');
4 app.use(express.static(path.join(process.cwd(), 'source')));
5 console.log(process.env.PORT || 5000);
6 app.listen(process.env.PORT || 5000);
```

Рисунок 5.7.1 – Код web.js

На рисунку 5.7.2 відображен код package.json

```
1 {
2   "name": "DistStudyBot",
3   "version": "1.0",
4   "dependencies": {
5     "express": "3.1.x"
6   },
7   "engines": {
8     "node": "0.10.x",
9     "npm": "1.2.x"
10  }
11 }
```

Рисунок 5.7.1 – Код ge.json

Також необхідно додати профайл Heroku із наступним змістом: «web: node web.js»

Для розгортання бота на сервері Heroku необхідно зареєструватись на платформі завантажити проект на GitHub та встановити систему контролю версіями – Git. Потім копіювати репозиторій GitHub командою «git clone». Потім за допомогою команд «heroku create» та «git push heroku» створити сервер та завантажити в нього репозиторій відповідно.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

## 5.8. Висновки до розділу

В розділі був проведений аналіз та опис використаних інструментів створення бота та необхідного програмного забезпечення. Також були приведені приклади та інструкції використання.

В процесі розробки бота були використані: мова JavaScript для написання коду бота, інтегровані заздалегідь модулі Telegram Bot API та TeleBot, фреймворк Express.js для серверної частини, СУБД MySQL та мову SQL для створення бази даних. Останнім кроком бот був завантажений на хмарний сервер для автономності.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		



## 6 ОПИС РЕАЛІЗАЦІЇ

Основна мета боту – автоматизація взаємодії між користувачем та базою даних, в якій знаходяться потрібні предмети та матеріали з предметів. При старті роботи бот запитує користувача його ім'я та навчальну групу. Користувач вводить свої дані і бот починає пошук у базі на наявність групи у відповідній таблиці. Якщо пошук дав результат, бот реєструє користувача та вносить його ім'я та ідентифікатор у таблицю «Студенти». Також бот закріплює у таблиці ідентифікатор групи, яку ввів студент. Із таблиці «Групи» витягується ідентифікатор групи, а також лист посилань на чати для спілкування між студентами щодо предметів або завдання з них.

Далі бот перевіряє таблицю «Предмети». До кожного предмету за допомогою ідентифікаційного коду закріплені групи, які за навчальним планом вивчають цей предмет. Бот витягує з таблиці лист предметів, ідентифікатори предметів, а також посилання на канали для лекційних занять та чати для практичних занять.

Наступним кроком за допомогою ідентифікатору предмету проводиться пошук у таблицях «Викладачі» та «Матеріали». З першої витягується лист викладачів, що закріплені за предметом та їхні контакти для зв'язку. З другої таблиці отримуємо усі доступні у реальний час документи, методички, презентації, лабораторні роботи та, якщо існують, дедлайни до виконання робіт у матеріалах.

У результаті бот збирає усю цю інформацію та відправляє студенту, що зареєструвався до групи за допомогою бота. Користувач отримує:

- лист предметів, що вивчає його група протягом навчального семестру ;
- лист викладачів з кожного предмету ;
- контакти викладачів для індивідуального зв'язку при виникненні питань або будь-яких проблем ;
- посилання на чати для спілкування з групою ;
- посилання до лекційних каналів з кожного предмету і чати з практичними заняттями ;

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

- усі доступні матеріали з кожного предмету ;
- інформацію щодо строків виконання робіт.

Отже студент може розпочати навчання одразу після отримання цієї інформації.

Після того як бот відправив студенту все необхідне, взаємодія з ботом на деякий час завершується. Періодично проводитиметься перевірка на наявність нових матеріалів та дедлайнів. Якщо є нові доступні файли, бот повідомить про це студента, витягне із бази та відправить їх студентові, як і інформацію про дедлайни.

Таким чином бот надає студенту усі необхідні інструкції для виконання навчального плану, а також спрощує пошук за групами, предметами, чатами для нових користувачів.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

## 7 ТЕСТУВАННЯ

Телеграм доступний у формі додатку для смартфонів, пристроях із операційними системами Android, IOS або macOS, Windows та Linux. Для тестування необхідно використати стаціонарний комп'ютер або ноутбук та смартфон на базі одної з операційних систем, що підтримують Телеграм.

Тестування проведено на смартфоні iPhone 7 із технічними характеристиками:

Екран: 4,7 ", IPS, 1334x750, мультитач;

Процесор: Apple A10 Fusion;

Графічний прискорювач: PowerVR GT7600;

Операційна система: iOS v 13.1;

Оперативна пам'ять: 2 ГБ;

Вбудована пам'ять: 64 ГБ;

Акумулятор: 1960 мАг;

Габарити: 138.3x67.1,x7,1 мм;

Telegram: Telegram v 6.2.

Також із використанням ноутбука з технічними характеристиками:

Відеоадаптер: GeForce GTX 1650;

Процесор: Intel Core i7-9750H CPU 2.60GHz;

Операційна система: Microsoft Windows 10 Ultimate;

Оперативна пам'ять: 8 ГБ;

Вбудована пам'ять: 256 ГБ;

Telegram: Telegram desktop v 2.1.3.

Для тестування бота необхідно його знайти за допомогою унікальної адреси - @diststudy\_bot. Далі бот запитує ім'я користувача. Швидкість відправлення повідомлень зумовлена стабільністю серверів Телеграма та інтернету на пристрої. На рисунку 7.1 зображений результат тестування на персональному комп'ютері із використанням Телеграму для Windows 10 версії 2.1.3.

Зм.	Лист	№ докум.	Підпис	Дата		Лист

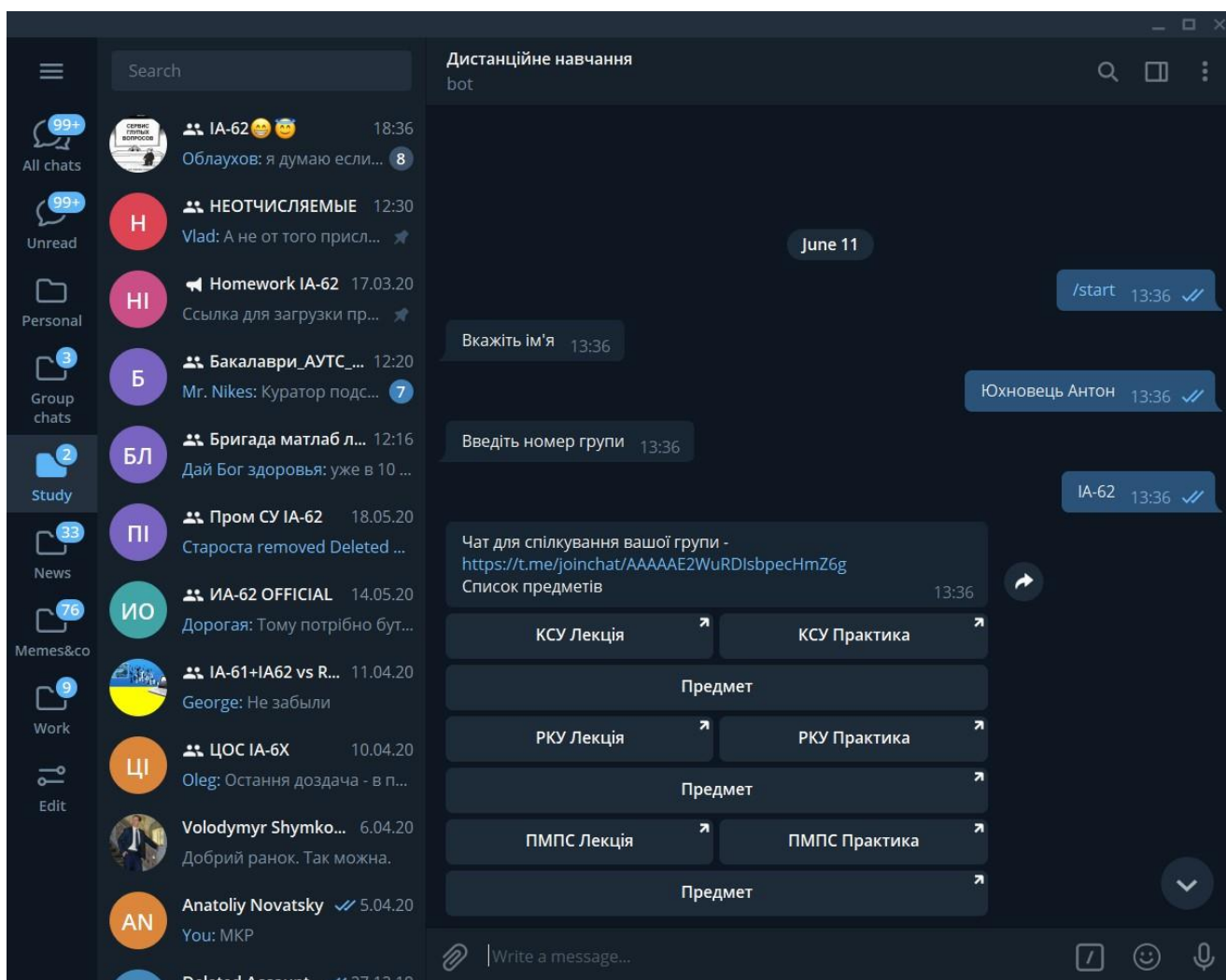


Рисунок 7.1 – Тестування у Telegram desktop

Автономність бота зумовлена доступністю телеграму на пристроях, на яких не встановлений додаток. Користування ботом може проходити за допомогою браузерної версії. На рисунку 7.2 зображений результат тестування на персональному комп'ютері із використанням браузера Opera GX та Web версії Телеграма.

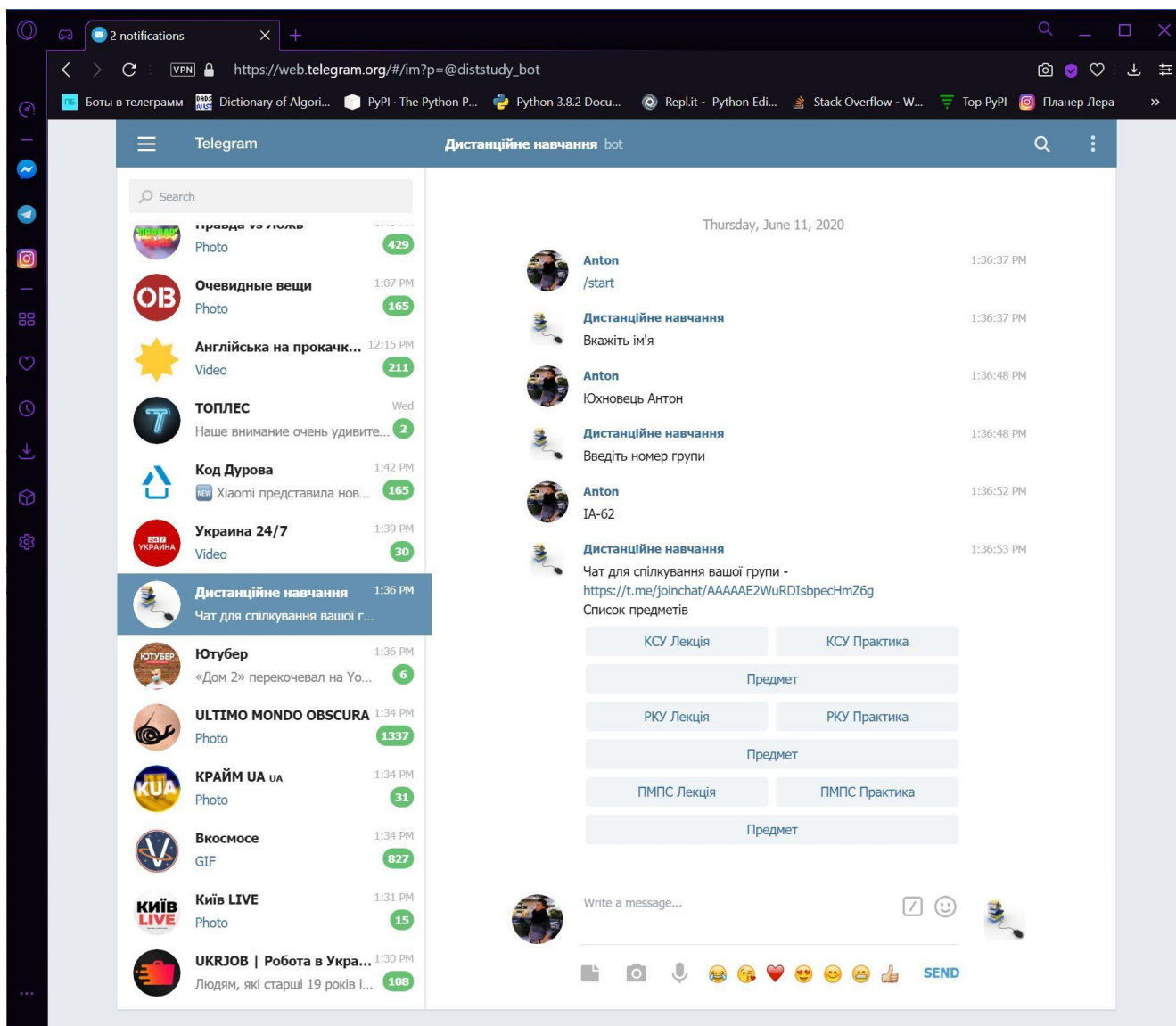


Рисунок 7.2 – Тестування у Telegram Web

Найбільш розповсюджений додаток Телеграм на мобільних пристроях, що також є важливим критерієм автономності. Наразі актуальна версія 6.2 для смартфонів із операційною системою iOS. На рисунку 7.3 зображений результат тестування на смартфоні із використанням Телеграму версії 6.2.

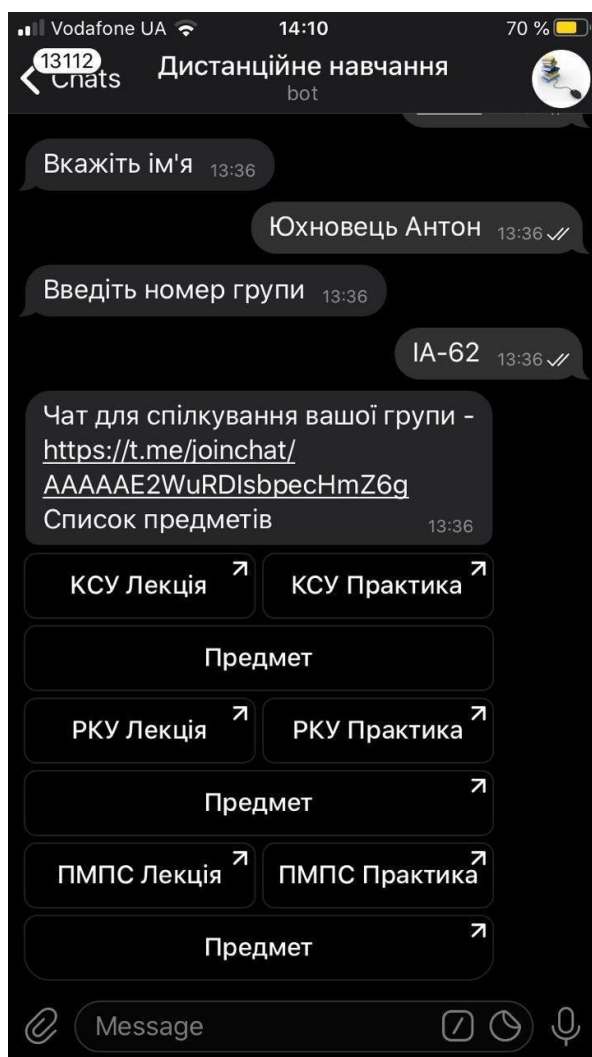


Рисунок 7.3 – Тестування на смартфоні з Telegram v 6.2

З проведеного тестування зроблено висновок, що створений бот працює виправно. Швидкість відповіді на запит користувача миттєва при стабільній роботі серверів Телеграму.

## ВИСНОВКИ

В проєкті був створений інструмент у форматі чат-бота на основі месенджеру Телеграм для проведення дистанційного навчання студентів. Був проведений огляд існуючих рішень та порівняльний аналіз обраного метода реалізації з доступними аналогами та встановлено переваги, що виділяють платформу та інструменти і завдяки яким є доцільним використання Телеграму задля створення бота.

Були встановлені вимоги, яким має відповідати інструмент, зокрема:

- можливість комунікації студента і викладача ;
- можливість проведення лекційних та практичних занять ;
- можливість проведення оцінювання ;
- вільний доступ студента до навчальних матеріалів ;
- можливість контролю доступу до навчального процесу.

В ході розробки програмного забезпечення проведений аналіз інструментів доступних до використання, зроблений опис процесу розробки. Наведено команди та методи за допомогою яких створений бот, алгоритми створення та взаємодії з базами даних, необхідних для реалізації поставленої задачі. Опис створення та роботи серверної частини, на яку завантажена база.

На виході отриманий бот, що має функції:

- ідентифікація студента та внесення у базу даних нових користувачів із прив'язкою до групи ;
- надання необхідних посилань для зв'язку із викладачем, групою, для участі у лекційних на практичних заняттях ;
- надання необхідної інформації щодо предметів, матеріалів та викладачів;
- відправка студенту навчальних матеріалів з предметів ;
- інформування щодо строків виконання робіт студентом ;
- моніторинг на з'явлення нових матеріалів та повідомлення щодо них і відправка студентам.

Зм.	Лист	№ докум.	Підпис	Дата		Лист

Бот автоматизує роботу студента з Телеграмом, також є автономним, простим та доступним у використанні.

Отже є всі підстави вважати мету проєкту досягнутою. Створене програмне забезпечення відповідає вимогам та цілям, а також має додатковий функціонал, що виражений у можливостях, які додають перевагу боту при виборі методик дистанційного навчання.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		



## ПЕРЕЛІК ПОСИЛАНЬ

1. Документація Telegram Bot API [Електронний ресурс] / Telegram – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.
2. Nemeth G. Node Hero - Node.js Security Tutorial / Gergely Nemeth.
3. Introducing JSON [Електронний ресурс] – Режим доступу до ресурсу: <https://www.json.org/json-en.html>.
4. Модуль Telebot для JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/mullwar/telebot>.
5. Приклади Telebot [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/package/telebot>.
6. Спосіб отримання даних об'єкта в окремому модулі [Електронний ресурс] – Режим доступу до ресурсу: <https://issue.life/questions/52320668>.
7. Методи HTTP запроса [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/HTTP/Methods>.
8. SQL запити [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/480838/>.
9. Документація Нероку [Електронний ресурс] – Режим доступу до ресурсу: <https://devcenter.heroku.com/categories/reference>.
10. Парсинг [Електронний ресурс] – Режим доступу до ресурсу: <https://ipipe.ru/info/parsing>.
11. HTTP [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/HTTP>.

Зм.	Лист	№ докум.	Підпис	Дата		Лист

12. How to build a Telegram Bot using Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://itnext.io/part-1-building-a-bookmark-manager-telegram-bot-node-js-vue-js-972bb7198def>.

13. Використання проміжних обробників [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/ru/guide/using-middleware.html>.

14. Маршрутизація в Express [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/ru/guide/routing.html>.

15. MYSQL — Базы даних [Електронний ресурс] – Режим доступу до ресурсу: <https://masterhost.ru/support/doc/mysql/>.

						Лист
Зм.	Лист	№ докум.	Підпис	Дата		

